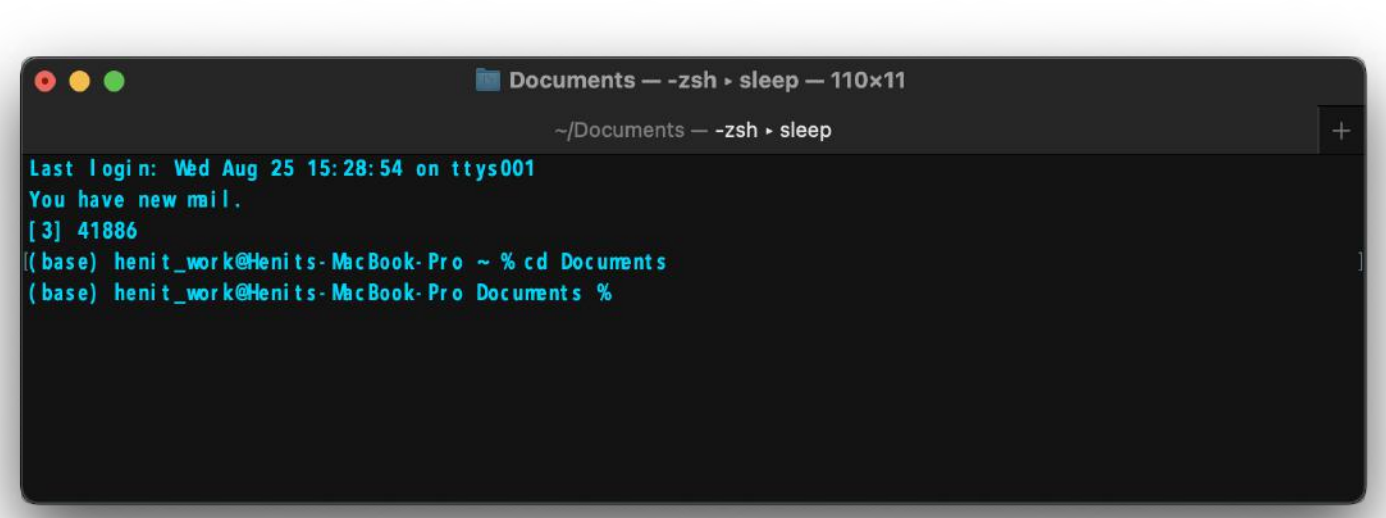
L-Digital Assignment - 1

> Henit Chobisa
> 20BDS0316
> C1 - TC1 Slot
> Jagalingam P Sir

Task 1 - Shell Commands

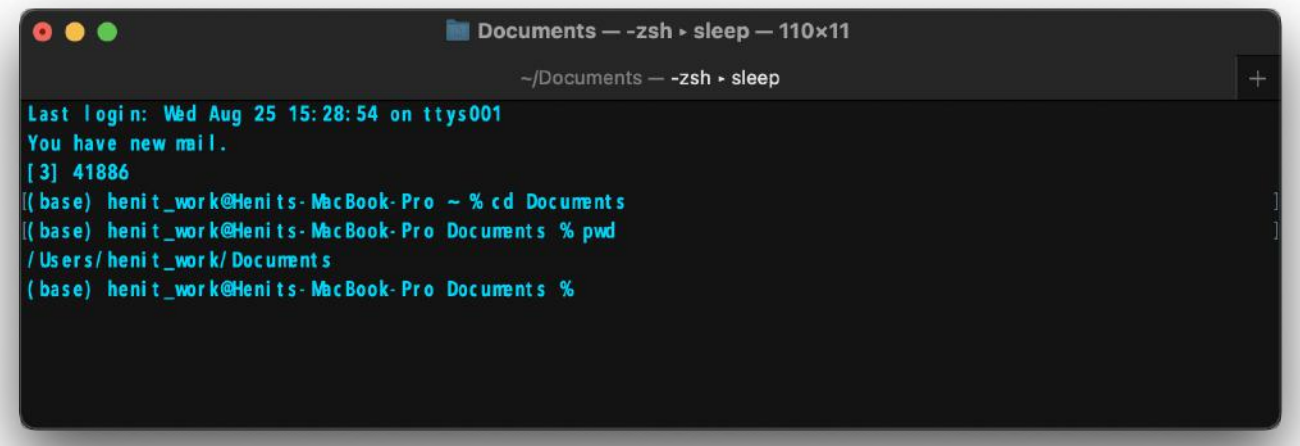
cd :

This commands helps us to change the current directory



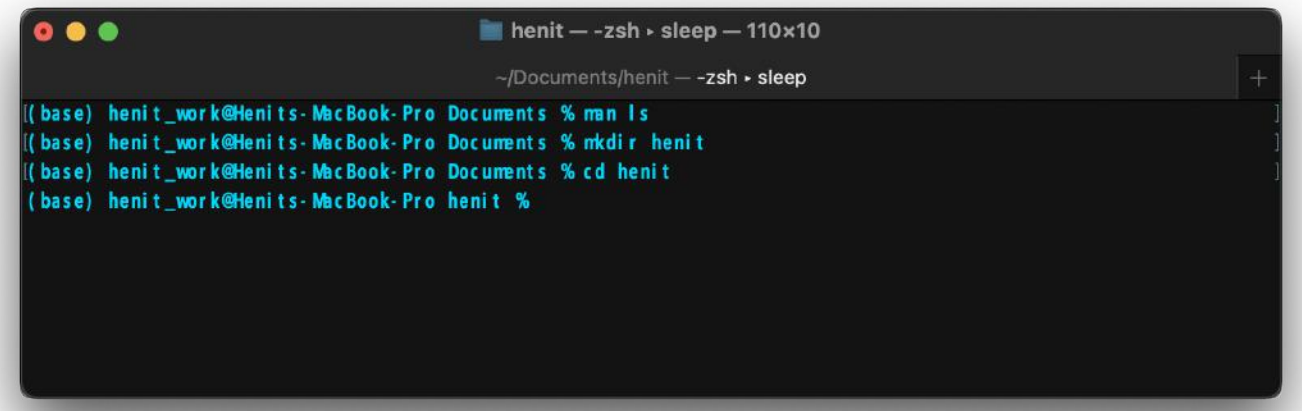
pwd :

this command helps us in determining the previous the current working directory of the shell.



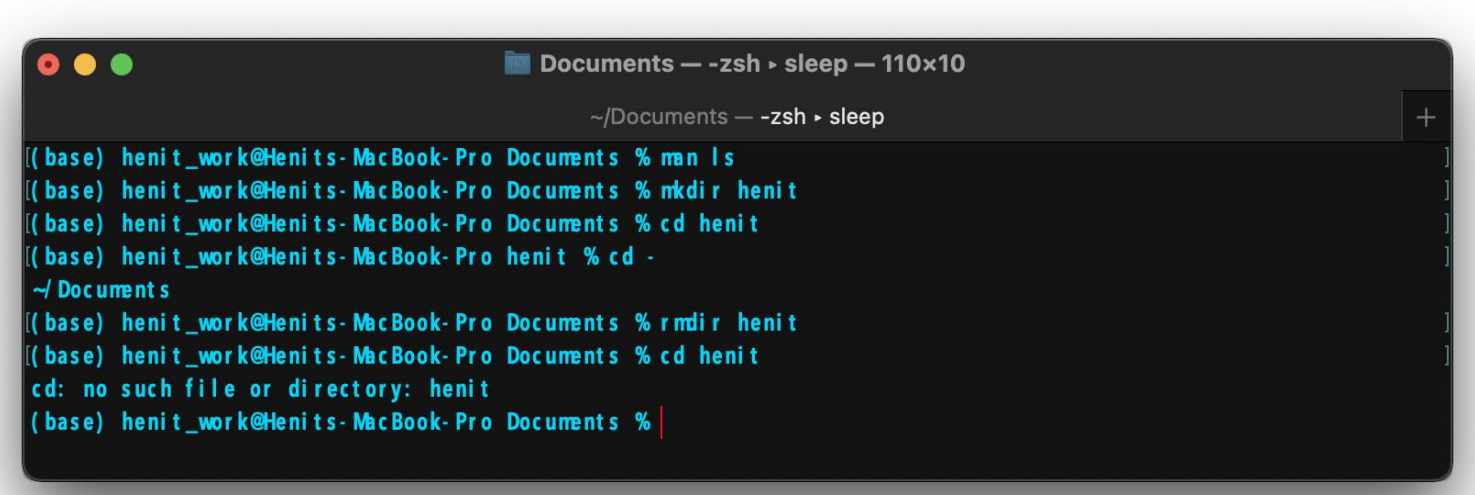
mkdir :

This commands, represents make directory, this can also be used with cd as a followup, so that if the directory is not present, it will create the directory and change the current directory to that particular.



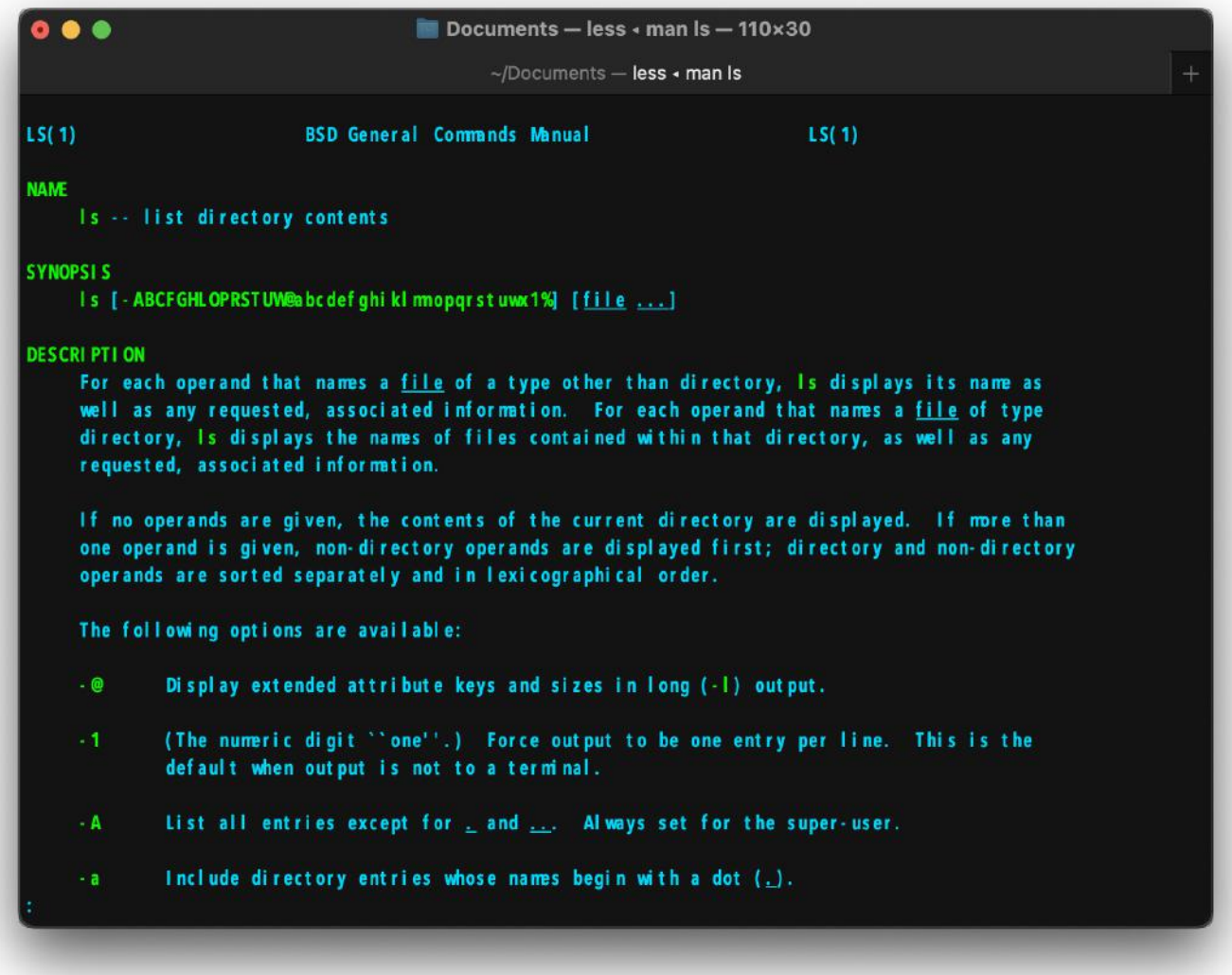
rmdir :

This commands helps in deleating a particular directory from the current directory.



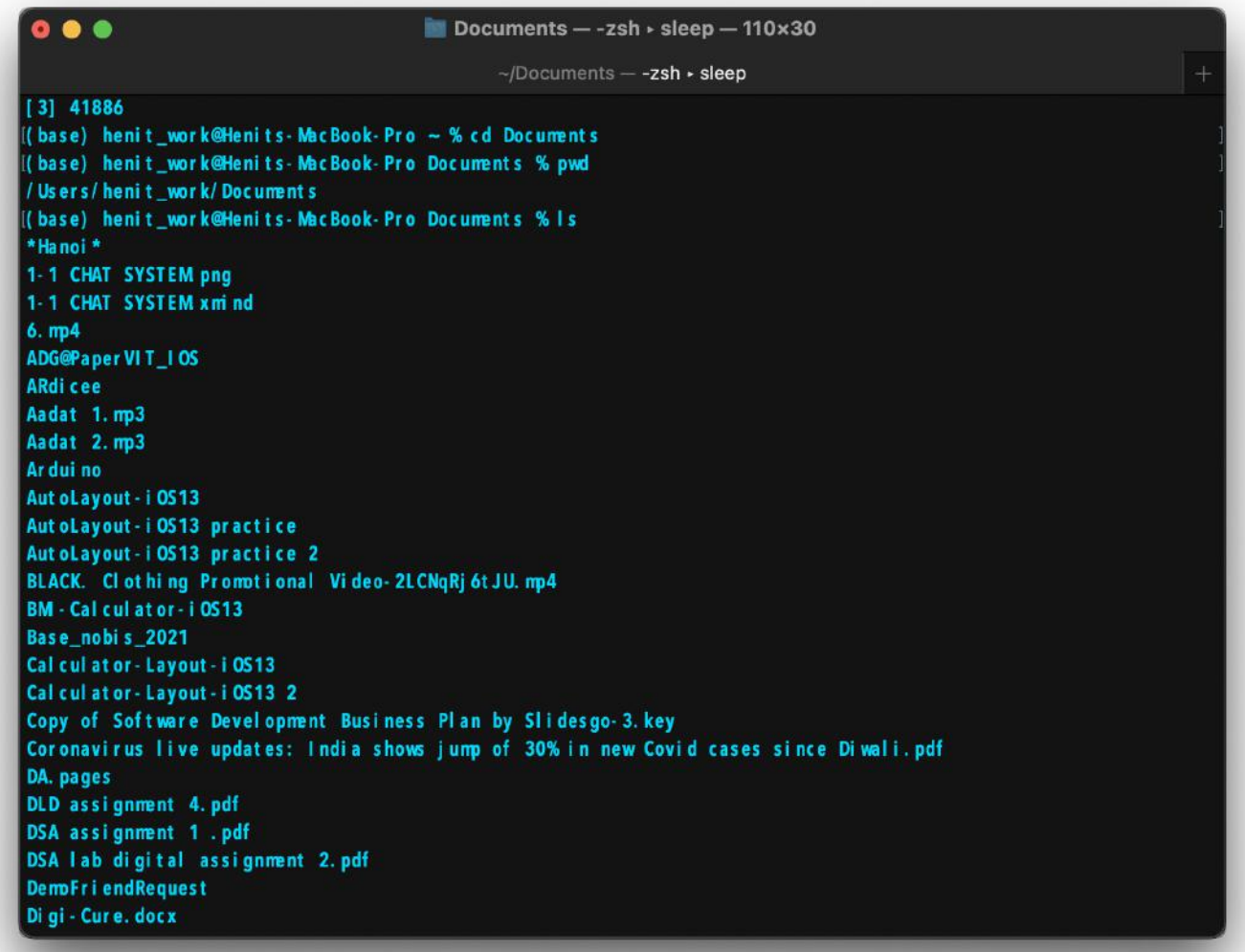
man :

man represents manual, it helps us to get a brief information about any specific command.



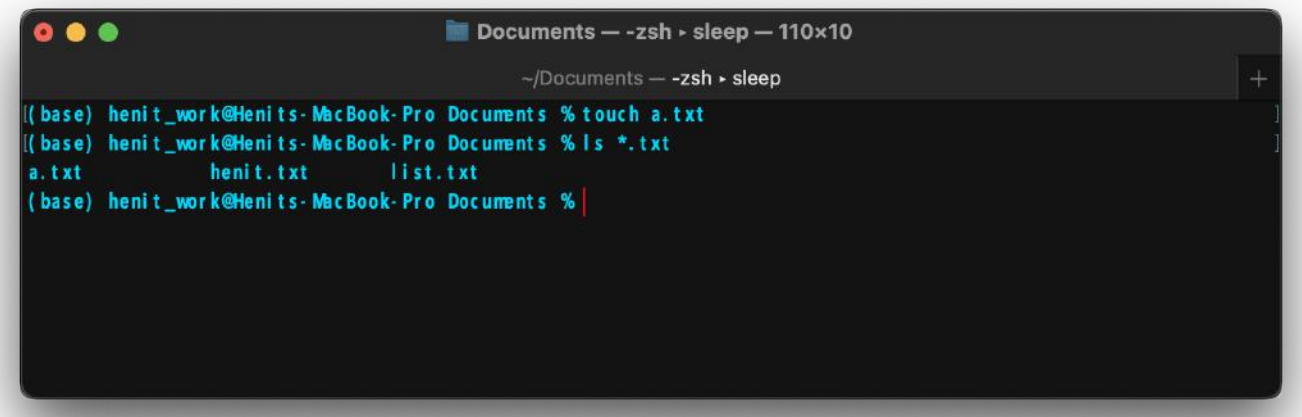
ls :

ls or list command lists out all the files in the current directory.



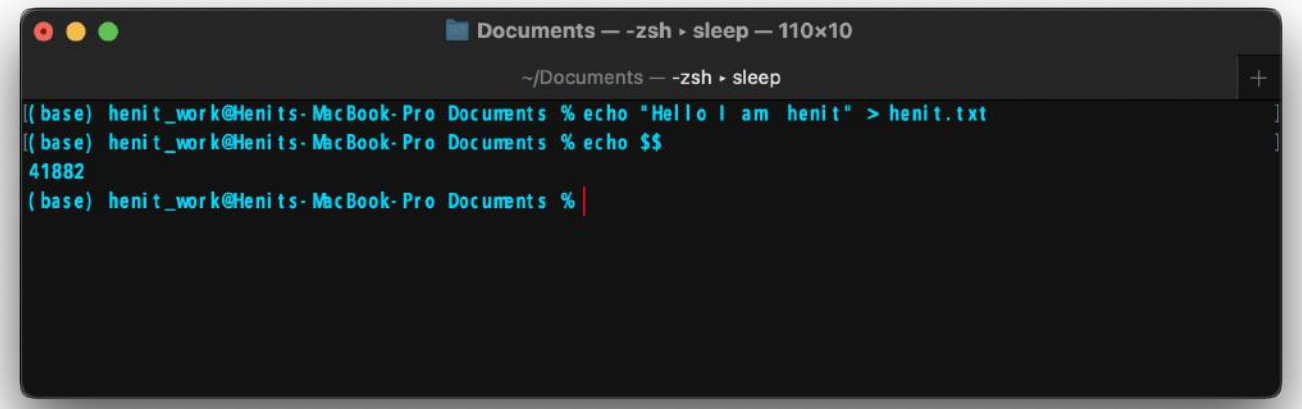
touch :

Touch helps us to create a file from the terminal itself, touch file name



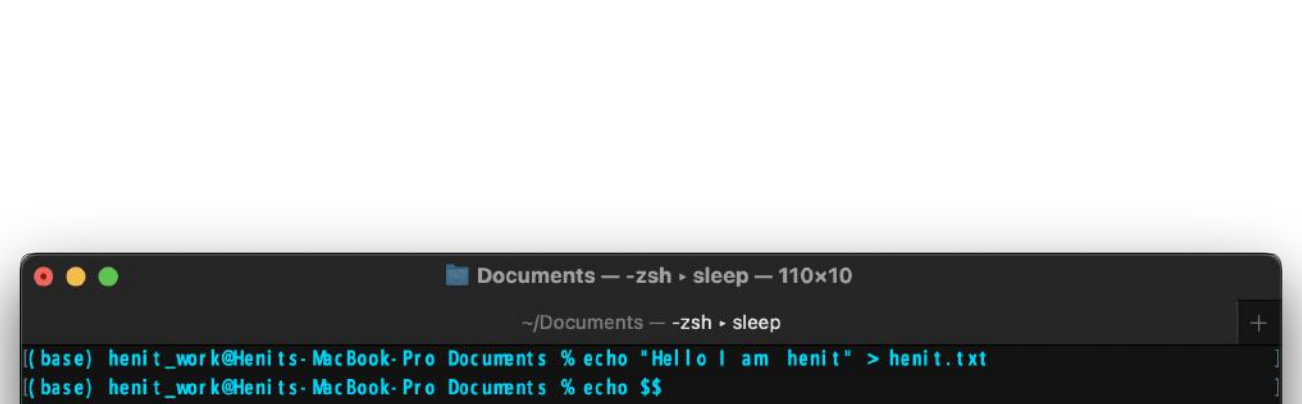
echo :

echo works as a print statement in shell, echo $$ gets you the process id of the shell



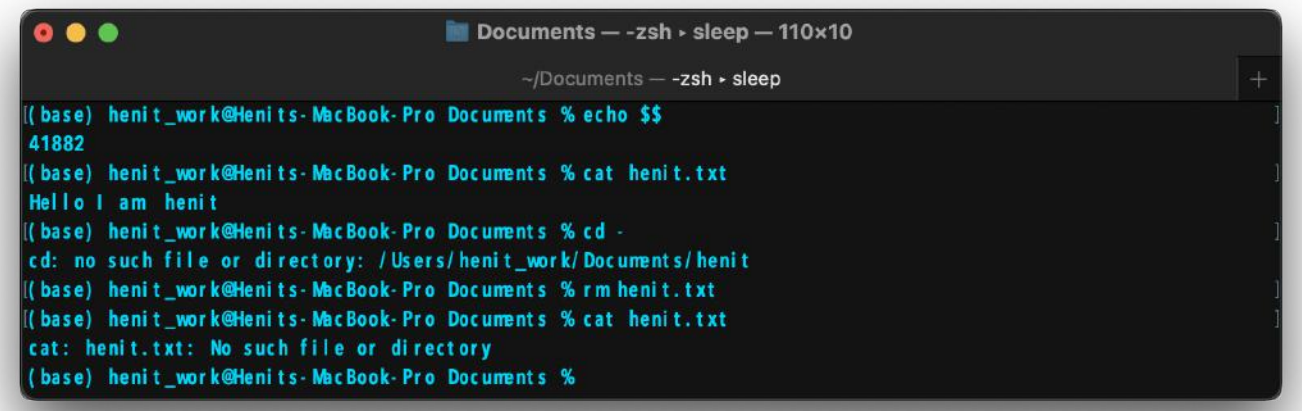
cat :

cat helps us to view the files in terminal itself.



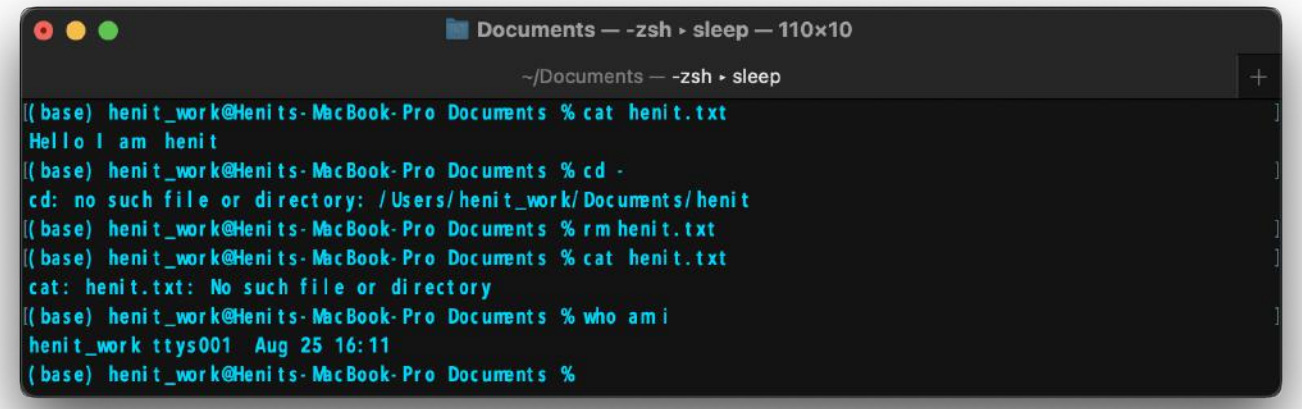
rm :

This command followed by the name of the file, helps us to delete a particular file.



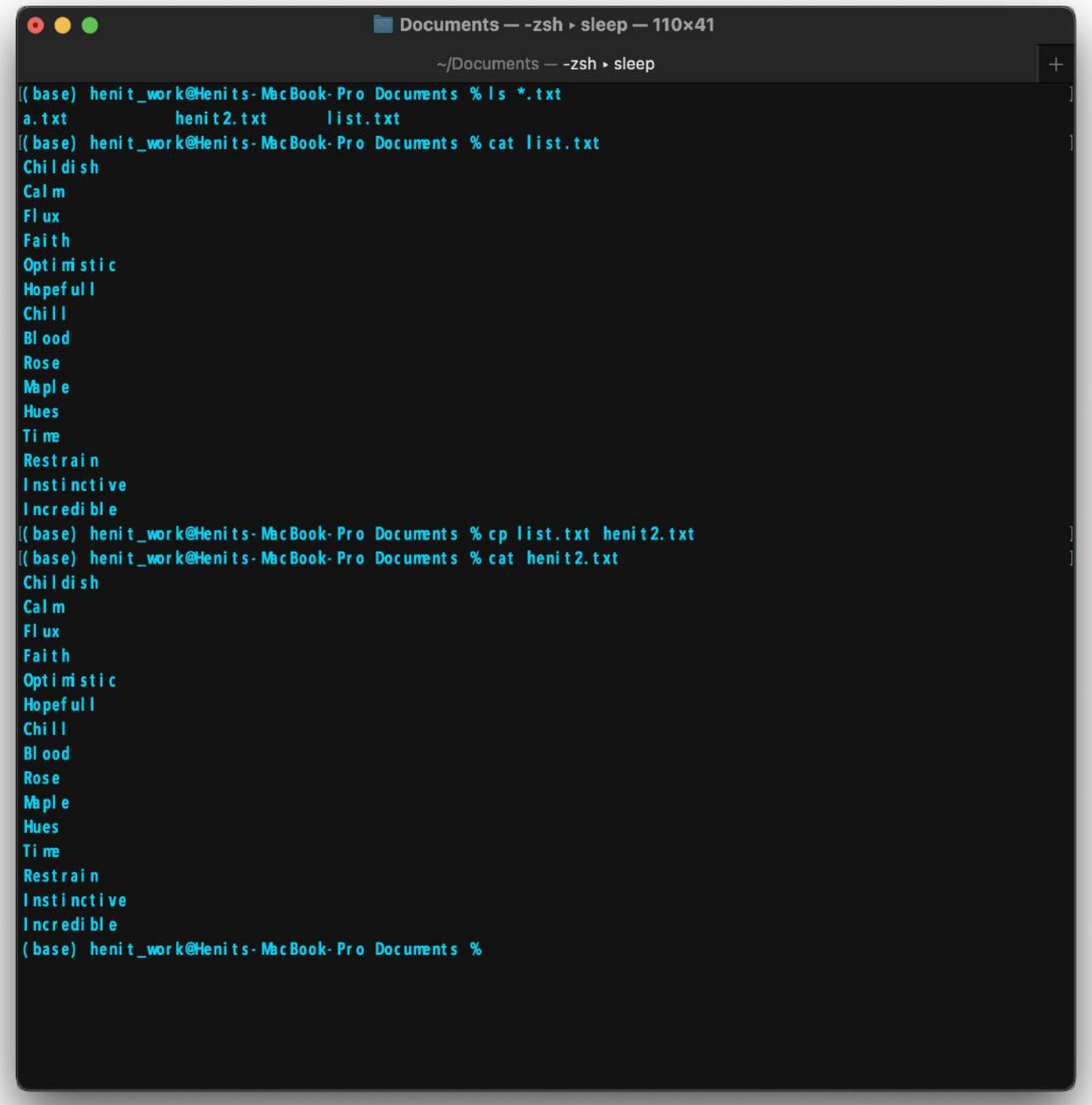
whoami :

whoami, helps us to know the administrator of the computer.



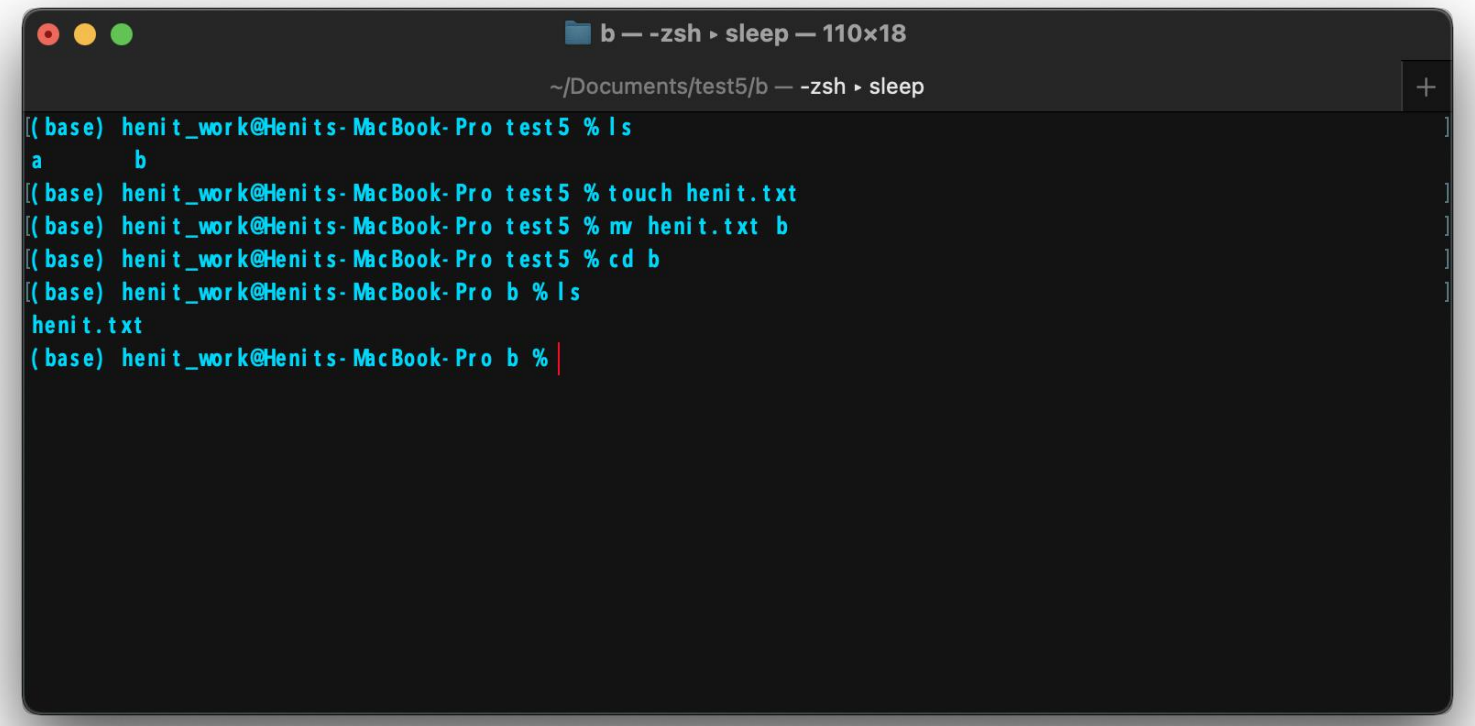
cp :

cp command help us to copy a file, cp <file 1> <file 2>



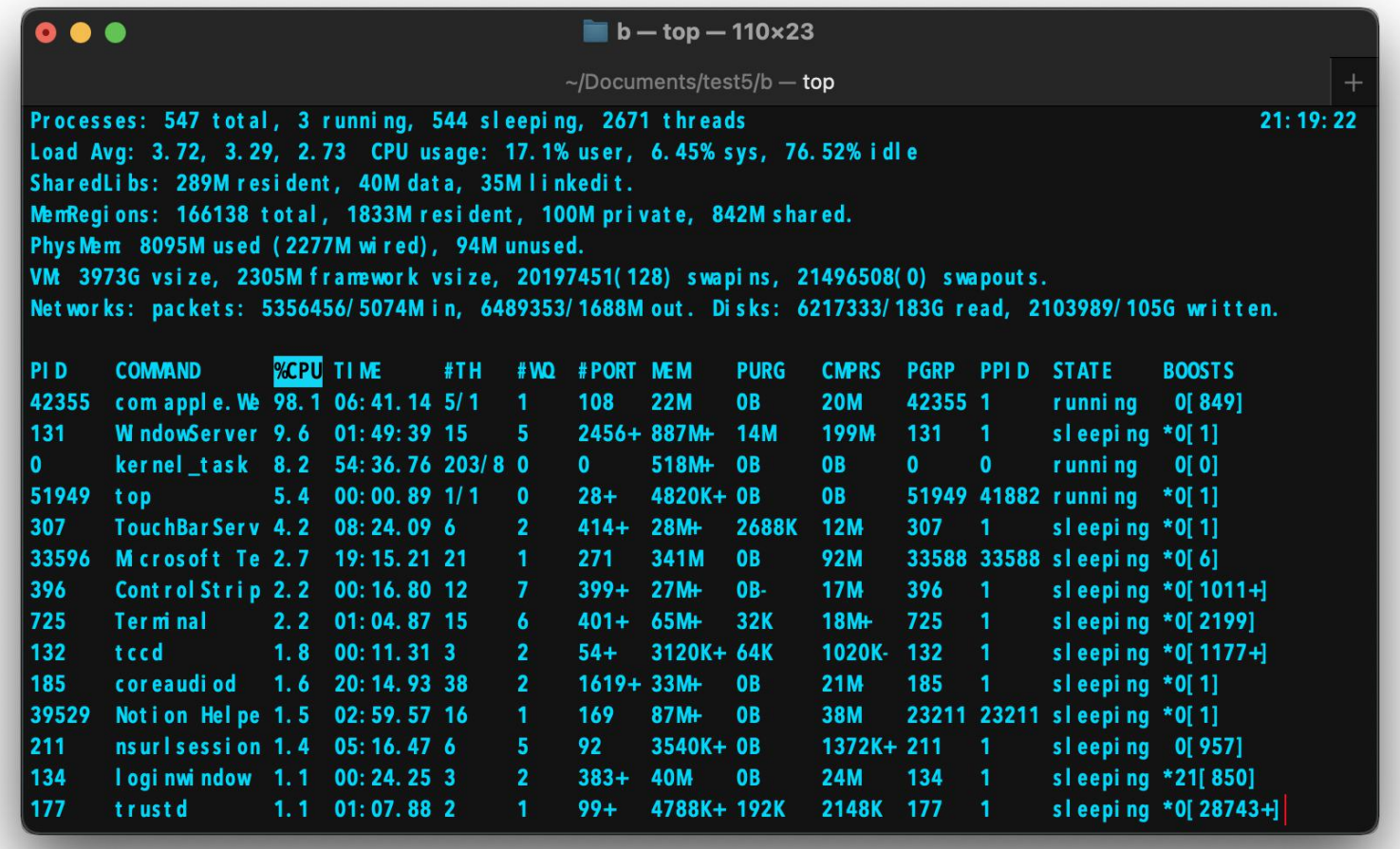
mv :

mv is a respresentation of move, it helps us to move a file from one directory to another.



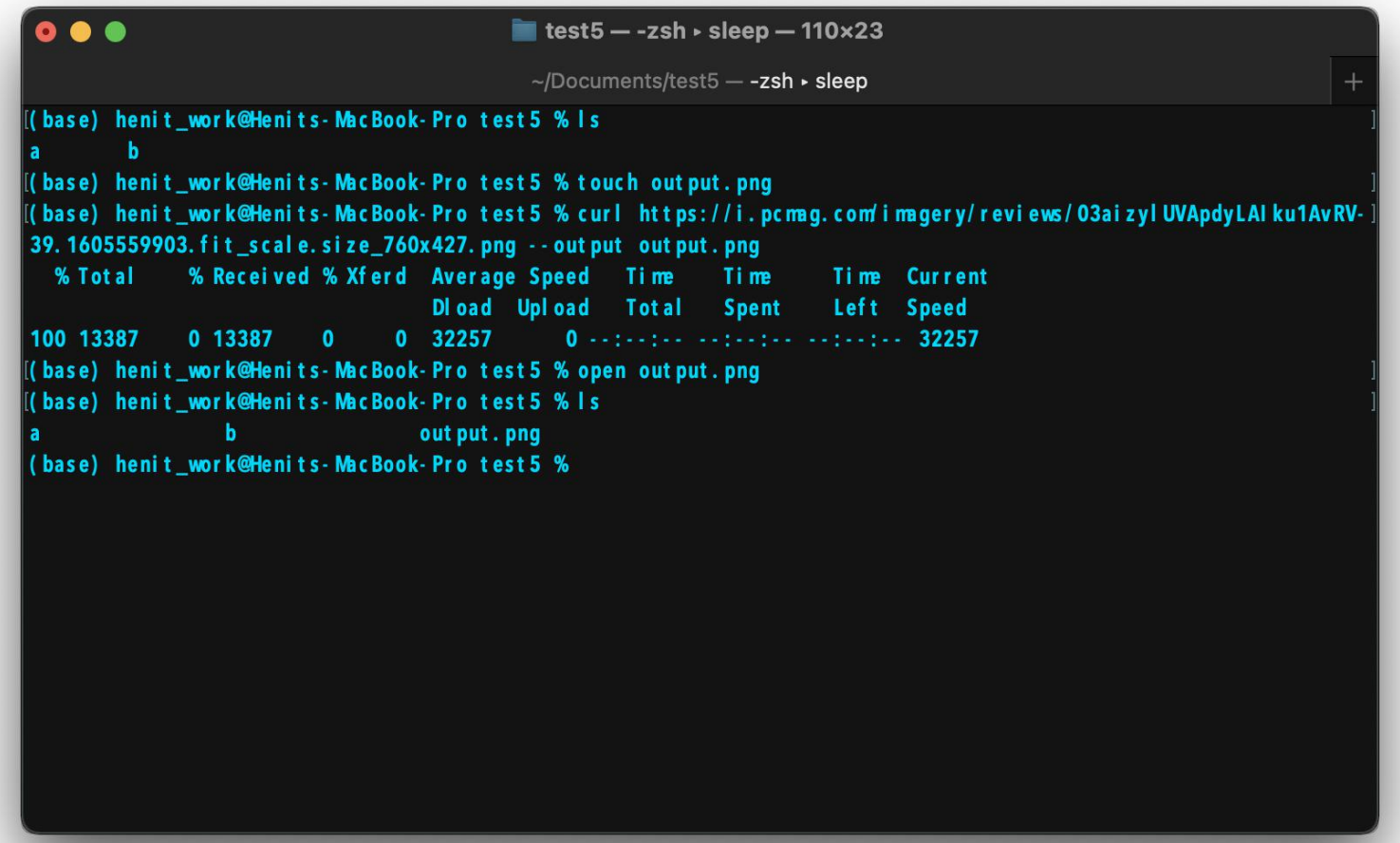
top :

top command list out all the current processes running in the system



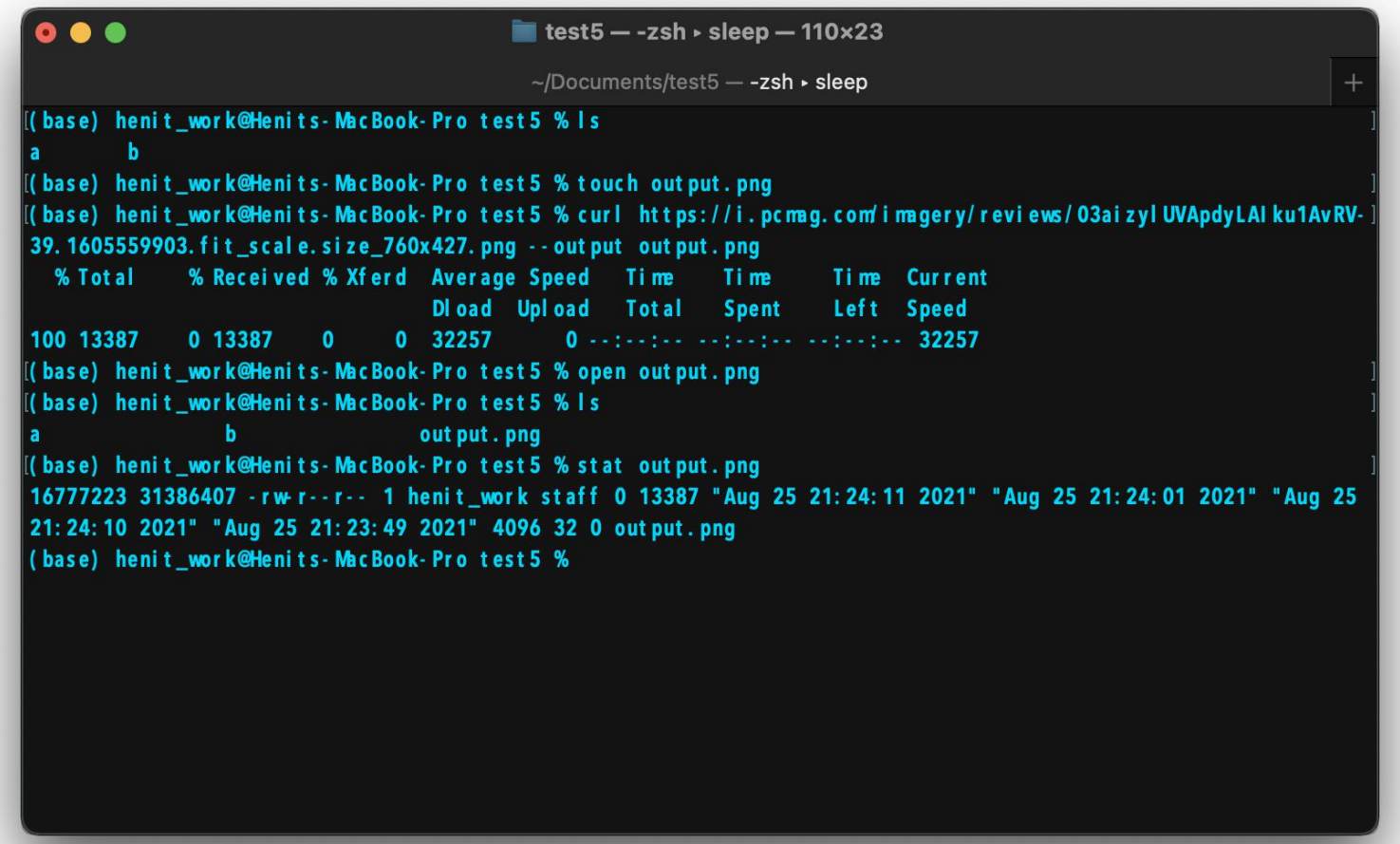
curl :

curl helps us to download a particular file from the internet to our pwd.



stat :

stat represents status, stat followed by file name will provide the current status of a particular file.



wc :

wc represents wordcount, execution of this command with a file name, will give the number of lines followed by number of words followed by number of letters.

```
test5 — -zsh ▸ sleep — 110×23
~/Documents/test5 — -zsh ▸ sleep
(base) henit_work@Henits-MacBook-Pro test5 % ls
a       b       i.txt
(base) henit_work@Henits-MacBook-Pro test5 % cat i.txt
I am an IOS Developer
(base) henit_work@Henits-MacBook-Pro test5 % wc i.txt
       1       5      23 i.txt
(base) henit_work@Henits-MacBook-Pro test5 %
```

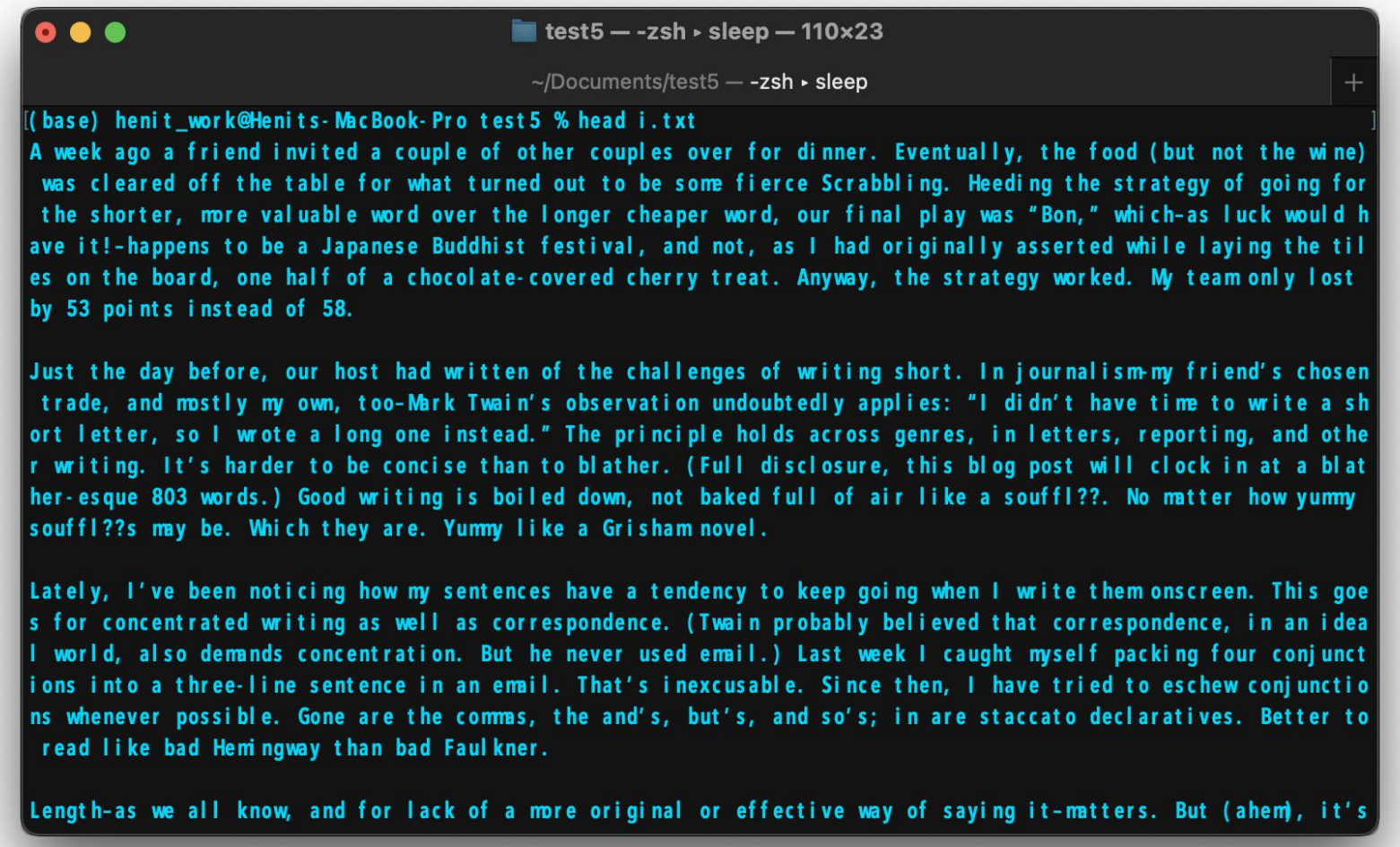
history :

history lists out all the commands previously executed.

```
test5 — -zsh ▸ sleep — 110×23
~/Documents/test5 — -zsh ▸ sleep
a       b       i.txt
(base) henit_work@Henits-MacBook-Pro test5 % cat i.txt
I am an IOS Developer
(base) henit_work@Henits-MacBook-Pro test5 % wc i.txt
       1       5      23 i.txt
(base) henit_work@Henits-MacBook-Pro test5 % history
 1085  clear
 1086  ls
 1087  cat "I am an IOS Developer" i.txt
 1088  clear
 1089  touch i.txt
 1090  cat "I am an IOS Developer" i.txt
 1091  clear
 1092  ls
 1093  cat i.txt
 1094  cat "I am an IOS Developer" > i.txt
 1095  echo "I am an IOS Developer" > i.txt
 1096  wc
 1097  clear
 1098  ls
 1099  cat i.txt
 1100  wc i.txt
(base) henit_work@Henits-MacBook-Pro test5 %
```

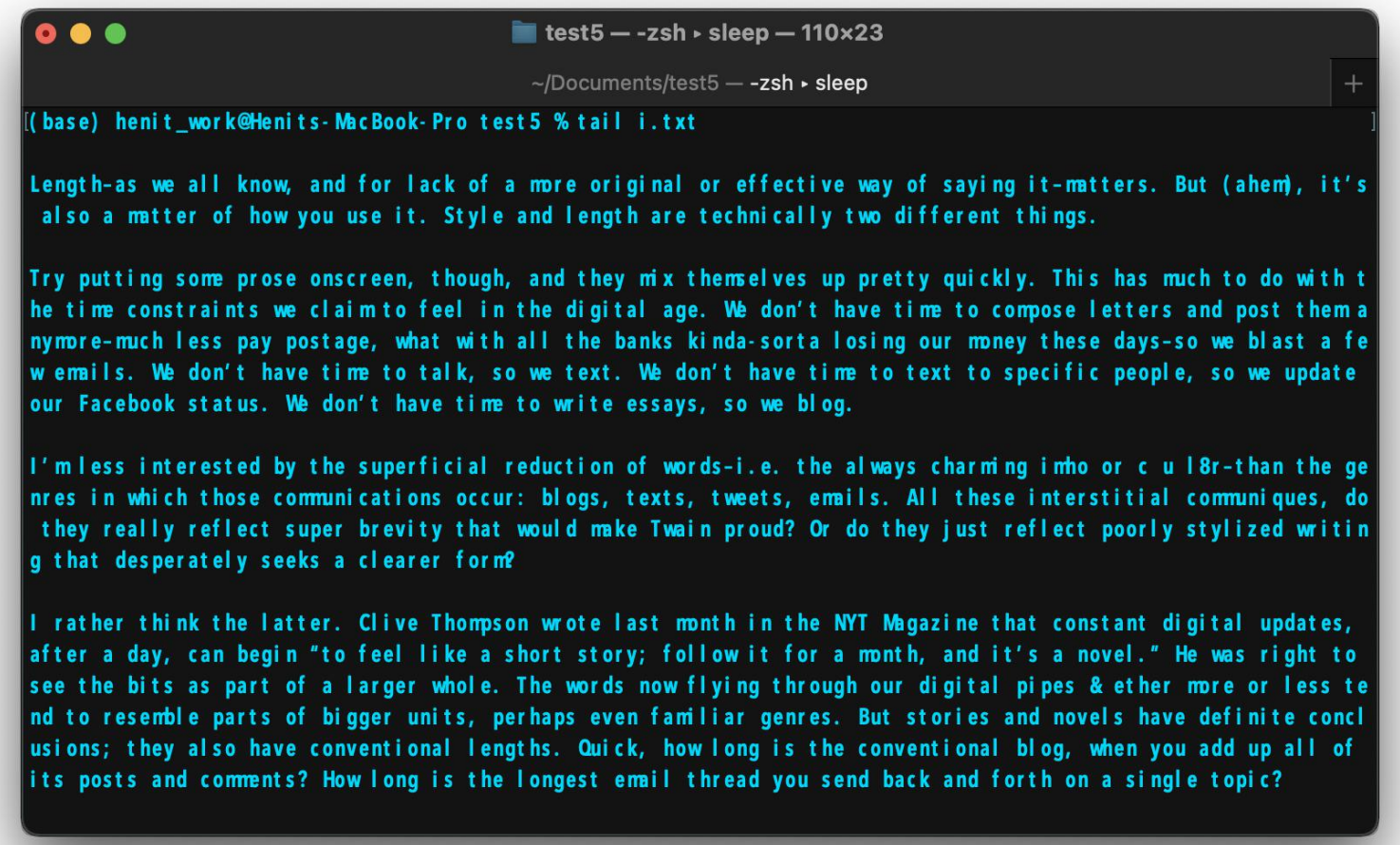
head :

head prints the initail paragraph of a few lines of a particular files.



tail :

tail prints couple of lines from the last part of file.



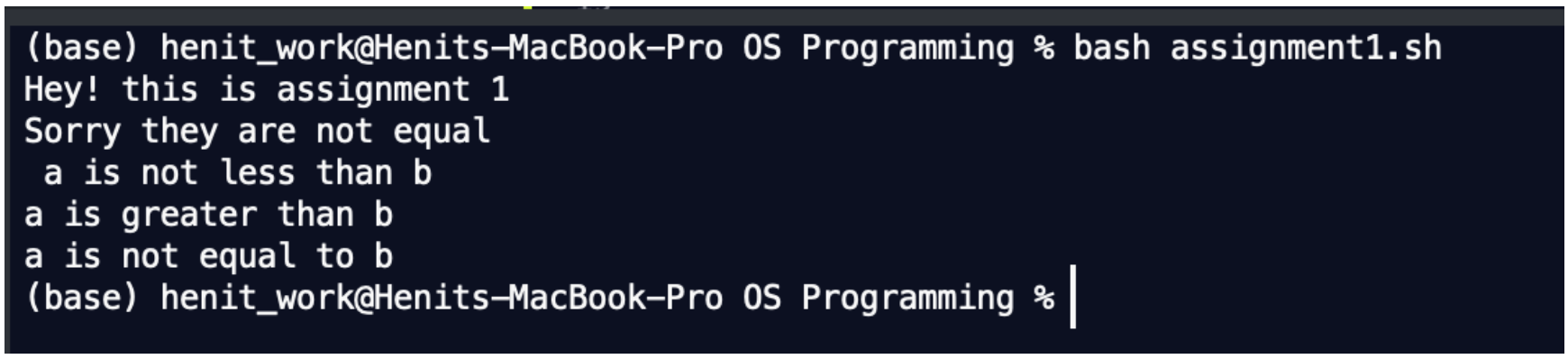
Task 2 - Shell Scripts :

Question 1 :

Consider two operands a and b which has value of 20 and 10 respectively write shell scripts to execute the relation operators such as -eq , -ne, -gt, -le.

```
echo "Hey! this is assignment 1"

a=20
b=10

if [ $a -eq $b ]
then
    echo "They both are equal"
else
    echo "Sorry they are not equal"
fi

if [ $a -le $b ]
then
    echo "a is less that b"
else
    echo " a is not less than b"
fi

if [ $a -gt $b ]
then
    echo "a is greater than b"
else
    echo "a is not greater than b"
fi

if [ $a -ne $b ]
then
    echo "a is not equal to b"
else
    echo " a is equal to b "
fi
```

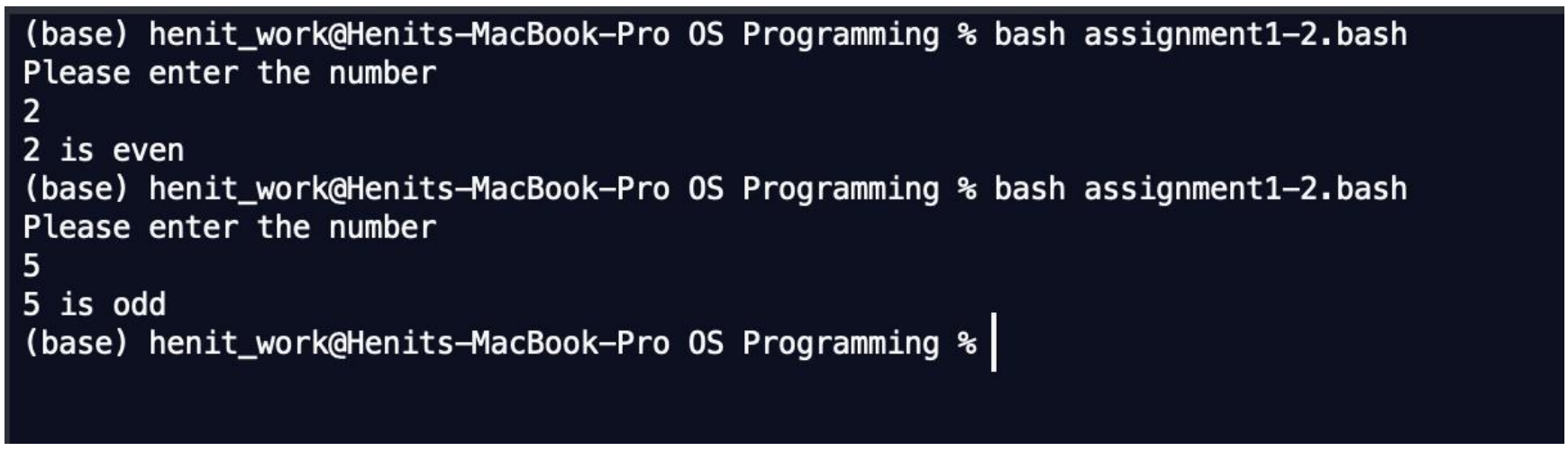
Output :



Ques 2

Shell script to find the number is even or odd

```
echo "Please enter the number"
read n

if [ `expr $n % 2` == 0  ]
then
    echo "$n is even"
else
    echo "$n is odd"
fi
```



Ques 3

Write shell script to find the largest number among the five

```
i=1
max=0

echo "Enter Numbers"
while [ $i -le 5 ]
do
  read num
  if [ $i -eq 1 ]  #set first number as max
  then
      max=$num
  else             #from number 2 update max if the num > max.
      if [ $num -gt $max ]
      then
        max=$num
      fi
  fi
  i=$((i + 1))  #increment i by 1
done

echo $max
```

```
(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1-3.sh
Enter Numbers
1
2
3
4
5
5
(base) henit_work@Henits-MacBook-Pro OS Programming %
```

Ques 4

Write a shell script to find the largest element in the array

```
echo "Enter Size(N)"
read N

i=1
max=0

echo "Enter Numbers"
while [ $i -le $N ]
do
  read num
  if [ $i -eq 1 ]  #set first number as max
  then
      max=$num
  else             #from number 2 update max if the num > max.
      if [ $num -gt $max ]
      then
        max=$num
      fi
  fi
  i=$((i + 1))  #increment i by 1
done

echo $max
```

```
(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1-3.sh
Enter Size(N)
10
Enter Numbers
20
13
14
18
12
14
12
10
11
15
20
(base) henit_work@Henits-MacBook-Pro OS Programming %
```

Ques 5

Write a shell script to display element A to Z in a loop

```
chars=( {a..z} )
n=3
for ((i=0; i<n; i++))
do
  echo "${chars[i]}"
done
```

```
(base) henit_work@Henits-MacBook-Pro OS Programming % bash assignment1-3.sh
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z

(base) henit_work@Henits-MacBook-Pro OS Programming %
```